




Apparatus for enhanced tagged data processing in a generalized computer execution unit.

Patent Number:  EP0365186, A3
Publication date: 1990-04-25
Inventor(s): CIAVAGLIA STEPHEN J; YATES JOHN S JR
Applicant(s):: APOLLO COMPUTER (US)
Requested Patent:  JP2211535
Application Number: EP19890310269 19891006
Priority Number(s): US19880259792 19881019
IPC Classification: G06F9/30 ; G06F9/318 ; G06F9/44
EC Classification: G06F9/318D, G06F9/44L2
Equivalents:  US5045992

Abstract

A method and apparatus for improving the efficiency of executing arithmetic and logical operations on arguments provided by the execution of Lisp or similar languages, wherein the variable type is accompanied by a tag identifier. The present invention enhances the processing of tag related with the addition of two condition codes derived from the values of the pre-ALU arguments and the post-ALU result. The condition codes allow rapid determination of data types without additional execution cycle or hardware overhead resulting in the enhanced execution of the Lisp programs.

Data supplied from the **esp@cenet** database - I2

THIS PAGE BLANK (USPTO)

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平2-211535

Int. Cl.

G 06 F 9/44
9/38

識別記号

330 A
330

庁内整理番号

8724-5B
7361-5B

⑬ 公開 平成2年(1990)8月22日

審査請求 未請求 請求項の数 6 (全6頁)

⑭ 発明の名称 コンピュータプロセッサ実行装置

⑮ 特 願 平1-270138

⑯ 出 願 平1(1989)10月17日

優先権主張 ⑰ 1988年10月19日 ⑱ 米国(US) ⑲ 259792

⑳ 発 明 者 ジョン・エス・ヤツ アメリカ合衆国03062 ニューハンプシャー州 ナサ、シ
ツ、ジュニア ルバートン・ドライブ 16
㉑ 発 明 者 ステフェン・ジェイ・ アメリカ合衆国03062 ニューハンプシャー州 ナサ、マ
シャバグリア クドナルド・ドライブ 23
㉒ 出 願 人 アボゴ・コンピュータ アメリカ合衆国01824 マサチューセッツ州 シェルムス
ー・インコーポレーテッド フォード、ビレリカ・ロード 330
㉓ 代 理 人 弁理士 秋元 輝雄

明細書の浄書

明 細 書

1. 発明の名称

コンピュータプロセッサ実行装置

2. 特許請求の範囲

1. それぞれのタイプが明確に異なるビットパターンを有する少なくとも第1と、第2のデータタイプを混合したものが処理されるプロセッサで使用するためのプロセス実行装置において、

第1のオペランドと、第2のオペランドとを提供する第1のオペランドレジスタ及び第2のオペランドレジスタと；

プログラム命令を提供する手段と；

前記第1のオペランド及び第2のオペランドを受取り、それらから、前記第1のオペランド及び第2のオペランドと、プログラム命令とに従って結果を形成する演算処理装置と；

前記第1のオペランド及び第2のオペランドを受取るように接続されて、前記第1のオペランド

及び第2のオペランドの少なくとも一方のタイプを識別し、そのオペランドのタイプ識別にตอบสนองして条件コードを提供するゲート手段と；

前記条件コードと、前記プログラム命令とにตอบสนองする条件試験手段とを具備し、前記プログラム命令により規定されるプロセスが前記オペランドの少なくとも一方に対して不適正である場合、代替命令が前記演算処理装置に提供されるプロセス実行装置。

2. 前記データタイプはオペランドデータと、非オペランドデータとを含む請求項1記載のプロセス実行装置。

3. 前記整数データは特定の下位ビットにより識別され；

前記ゲート手段は、前記第1のオペランド及び第2のオペランドの前記特定の下位ビットを受取る請求項2記載のプロセス実行装置。

4. 前記整数データは2つの最下位ビットに関してゼロの値を有し；

前記ゲート手段は、2つのオペランドが共に整

数データであるときに第1の状態を有し、その他の場合には第2の状態を有する1ビットの条件コードを提供する請求項3記載のプロセス実行装置。

3. 第1のオペランド及び第2のオペランドを提供する手段と；

命令を提供する手段と；

前記第1のオペランド及び第2のオペランドと、前記命令とを受取り、それらから結果を提供する実行手段と；

前記結果の特定の低位ビットを特定するゲート手段とを具備し；

前記第1のオペランド及び第2のオペランドの最下位ビットはタグビットから成り；

前記ゲート手段の出力は前記タグビットの試験結果を提供する、ブランチ試験を実行するコンピュータシステム。

8. 前記第1のオペランド及び第2のオペランドは、最上位ビットと、最下位ビットとにそれぞれ関連する上部と、下部とをそれぞれ含む；

されるのではない。実行可能コードの中でデータのタイプを確定することができ、どのタイプのデータでも、変数に割当てることができる。タグ付けは、通常、タグ付きポインタを使用して実行される。Lispではポインタは非常に重要であるので、ポインタはできる限り経済的に使用されている。たとえば、ポインタにより表わされるアドレスと、ポインタの中でそのタイプを指示するビットとを組合せることにより節約ができる。LucidのCommon Lisp (LCL) や、Danaia Lisp (DL) においては、下位の3ビットをタグ用として留保し、全てのデータを8バイト境界に整列させ、且つ全ての関連アドレスに対する変位を利用してタグを「マスク」することにより、この経済的な利用を実現している。この方法はLCL、DL及びYaleのTで採用されており、汎用マシンにおけるポインタのタグ付け方法としては確實で効率良く、拡張可能であることがわかっている。

タグ付けを利用するということは、それに相応して、有効なビットフィールドマニプレータが必要

前記下部のうち少なくとも3つは、前記上部のデータタイプを示すタグデータを含む；

前記実行手段は、1つのオペランドタグの選択された第2のオペランドによる試験を実行し、そのタグと選択された第2のオペランドとが等価であることを示すゼロビット量が前記ゲート手段により検出され；

前記実行手段は、ゼロでないビット量の検出時に後続命令を受取って、新たな結果を提供するように動作する請求項6記載のコンピュータシステム。

3. 発明の詳細な説明

(産業上の利用分野)

本発明はコンピュータプロセッサ実行装置に関し、特に、対応するタグ情報を内包しているマルチタイプのデータのプログラムに従った処理に適合するコンピュータプロセッサ実行装置に関する。(発明の背景)

Lisp言語システムにおいては、データにタグが付されており、変数はコンパイル時にタイプ分け

要であることを示唆している。間接演算の一部としてタグをマスクすることは可能であるが、タイプを確認するため又はタイプをメッセージするためにはタグを検査しなければならず、効率の良いビットフィールド演算の必要性が確かめられる。

Lispの宇宙は原子と、リストセルとから構成される。この見方は意味論的には正しいモデル化であるが、実際にLispを適用する場合には、小さな変数についての演算はきわめて一般的であるので、それらは特殊なケースと考えられる。従って、最低レベルのインプリメンテーションでは、Lispの宇宙には実際には次の3種類の物体が存在している。

CONS-CELLS リストを構成しているポインタ2つのセル；

SMALL-INTS タグビットから取出されたポインタのサイズに関する小さな符号付き変数；及び

OTHER 拡張タグを有する記憶ブロック
3つの下位ビットタグ値への意味の割当ての具

型約な 1 例を次に示す。

- 000 30ビット偶数整数 (最上位タグビット
(-0) と結び付けられた29の上位ビ
ット)
- 001 ポインタ拡張 (タグ付き非演算記憶ブ
ロック)
- 010 数拡張 (浮動小数点又は無限精度固定小
数点)
- 011 Cons cell に対するポインタ
- 100 30ビット奇数整数 (最上位タグビット
(-1) と結び付けられた29の上位ビ
ット)
- 101 記号拡張 (特性リスト、値セルなど)
- 110 その他の即時拡張 (すなわち、非整数;
たとえば、単一文字)
- 111 使用せず (ハードウェアにより支援され
る場合は正方向ポインタ)

Lispは統計的にタイプ付けされておらず、また、
機械はタイプタグを検査することにより多形動作
するものに限定されているので、Lispコードの英

行時には次のシーケンスが繰返される。

機能入力

(このコードはインラインで非常に頻繁に拡張さ
れる)

引数のタグに関するケース

(いくつかのタグ値又はごく多数のタグ値でタイ
プ誤りを表わせる)

ケース解析に基づいて適切な演算を実行

理論上は、いずれの関数に関わるコードも、任
意のタイプの引数と対応するように、タイプディ
スパッチに基づいて異なるコードを実行するか、
あるいは、タイプ誤りを検出して通報することに
より、準備されなければならないが、ほぼ全ての
機能について、実際に発生するあらゆるケースの
うち圧倒的多数のものを1つの独特のタイプで
明示している。これは、CAR、CDR及び基本
演算機能の基本ケースに関して特に当てはまる。

(実 施 例)

以下、添付の図面を参照して本発明を詳細に説
明する。

本発明による装置及び方法は、1988年10月7日
出願の同時係属出願のAPOLL-105XX(名称「Method
and Apparatus for Concurrent Dispatch of In-
structions to Multiple Functional Units」)、
あるいは、一通の命令が記憶されている命令
キャッシュ102又はその他の記憶場所などの発生
源から、一般にはプログラムカウンタパイプラ
イン132と組合されているプログラムカウンタ又
はプログラムカウンタ発生器130により順次供給
されるアドレス信号にตอบสนองして提供され、一通の
命令を受取る実行装置110を有する他の公知のプ
ロセッサの中の1つに適用可能である。命令
キャッシュ102により提供される命令は、実行装
置110の一部であるレジスタ112へ直接発行され
ても良いが、従来の技術で知られている例に見ら
れるように、命令パイプライン104に一時的に留
まっても良い。第1図に示す実行装置110は、
A)オペランドレジスタ114と、B)オペランドレジ
スタ118と、レジスタ114及び118のオペランド
A)及びB)をそれぞれ受取る演算論理装置(ALU)

116とを含む。ALU116が提供した演算の結果
は、レジスタ120に記憶される。レジスタ120に
記憶された結果はレジスタファイル112へ送られ
た後、命令キャッシュ102から命令パイプライン
104を介して受取られた命令に従って、オペラン
ドレジスタ114,118の一方に選択的に提供され
ば良い。

本発明にも参考として取入れられている、
条件コードパイプライン122に関連する1988年10
月18日出願の同時係属出願 APOLL-107XX(名称
「Central Processor Condition Code and Appa-
ratus」)に説明されているように、演算論理装
置は、レジスタ120に記憶されている結果と一致
する4つのSTETビットから成る条件コード
(N, Z, V, C)を条件コードパイプライン
122に供給する。このパイプラインには条件コー
ドL2及びZ3も追加される。パイプラインの条
件コードの状態値はプログラムカウンタ発生器
130に供給され、プログラムカウンタ発生器130
は、条件コードパイプライン122により提供され

た条件コードと、レジスタファイル112により供給されるいずれかのブランチ宛先アドレス信号とに依存して、後続プログラムカウンタ用命令アドレスをPCパイプライン132へ供給する。

本発明の一面によれば、4入力ORゲートから構成される論理素子124は、それぞれのオペランド114及び116から提供されたオペランドの2つの最下位ビットを受取る。本発明の1つの適用用途によれば、オペランドレジスタ114及び116に記憶されているオペランドの3つの最下位ビットはタグ識別子ビットを構成し、オペランドの整数状態は、下位の2つのビットにあるゼロ値により示される。従って、AとBのオペランドレジスタ114及び116にそれぞれ存在するオペランドが共に整数値を構成する場合、最下位ビットはゼロ値を有するので、その結果、ORゲート124の出力端子の状態は論理値「0」となる。この論理値は条件コードパイプライン122に入力され、そこで、プログラムカウンタ発生器130は後続プログラムカウンタに符号をPCパイプライン132に供給す

る。Lispプログラムの中で数値データを処理するとき、オペランドレジスタ114及び116により供給される信号の大多数は整数値を構成するので、ORゲート124の出力は論理値「0」となり、L2はゼロの値を有する。相対的に見れば頻発ではないが、オペランドは数値有効範囲などの異なるタイプを成すときもあり、その場合には、ORゲート124の出力端子においてL2はゼロでない値を有する。本発明によれば、ゼロでない値が条件コードパイプライン122に取込まれて、プログラムカウンタ発生器130に入力されると、その結果、命令キャッシュ102により実行装置110に提供される命令の流れが変化して、オペランドレジスタ114及び116の一方又は双方で受取られる非整数データはその実際のタイプに従って処理されるようになる。

さらに詳細に言えば、本発明によるシステム装置及び方法は、図示のブロック線図50で示す構造により提供される、A〔0〕、A〔1〕、B〔0〕及びB〔1〕の論理和に等しいLisp特定L2条件

コードと、L2、L2、(L2+V)及び(L2+V)に等しい4つの新たなBR・COND術語とを追加するもので、その結果、整数に関する無作為演算のためのコードは、

第1のリスト

EXTRACT.ccRa.\$0.\$1.R31; Ra[1:0] = 00を検査
EXTRACT.ccRb.\$0.\$1.R31; Rb[1:0] = 00を検査
BEQ. sf 1\$

SVC #トラップ・コード; Ra→BIGNUM又はタイプ誤り

1\$ BEQ. sf 2\$

SVC #トラップ・コード; Rb→BIGNUM又はタイプ誤り

2\$ 演算. cc.Ra.Rb.Rc ;実際の演算を実行

BNY. sf 3\$

SVC #トラップ・コード; オーバフロー、BIGNUM発生

3\$

の内容から、次に示す例に縮小する。

第2のリスト

演算. cc.Ra.Rb.Rc ;実際の演算を実行

BIX. sf 1\$

SVC #トラップ・コード; オーバフローならば、BIGNUM発生

1\$; そうでなければ、タイプ誤りをトラップ

整数演算はライン外ルーチンへの呼出しにより処理されるので、実際の改善はこれらのコードフラグメントが示唆するよりさらに大きい。L2ビットと、新たなBR・COND術語との組合せにより、インライン整数コードの発生が可能になる。

本発明の別の面は、前述のOR論理ゲート124と一致して又はそれとは無関係に動作して、結果レジスタ120により供給される信号の3つの最下位ビットの補数の、AND論理装置126による3入力論理和組合せを得る。論理信号ラベルZ3を提供する論理装置126の出力は、結果信号の3つ

手続補正書 (方式)

平成2年2月 4日

特許庁長官 殿
(特許庁審査官 殿)

1. 事件の表示

平成1年 特許 第 270138 号

2. 発明の名称

コンピュータプロセッサ実行装置

3. 補正をする者

事件との関係 特許出願人

名称 アポロ・コンピューター・インコーポレーテッド

4. 代理人

住所 東京都港区南青山一丁目1番1号

〒107 電話 4761 1501

氏名 (6222) 井原士 秋 元 謙

住所 同 所

氏名 (1615) 井原士 秋 元 不二

5. 補正命令の日付 ()

(発送日) 平成2年1月30日

6. 補正の対象

願書、明細書全文、委任状 (訳文付)

7. 補正の内容

別紙のとおり

